AD 686293

IDENTIFICATION OF PARAMETERS BY THE METHOD OF QUASILINEARIZATION

CAL Report No. 164

14 May 1968

CORNELL AERONAUTICAL LABORATORY, INC.

OF CORNELL UNIVERSITY, BUFFALO, N. Y. 14221

CORNELL AERONAUTICAL LABORATORY, INC.
BUFFALO, NEW YORK 14221

CAL REPORT NO. 164

IDENTIFICATION OF PARAMETERS BY THE METHOD OF QUASILINEARIZATION

14 MAY 1968

This report is a detailed description of a numerical method that has been modified and programmed at CAL under internal research support.

PREPARED BY: Duane B. Larson
Duane B. Larson

APPROVED BY: John T. Fleck
John T. Fleck, Head
Computer Mathematics Dept.

John T. Fleck
John T. Fleck

ABSTRACT

The method of quasilinearization is a combination of the properties of the high-speed digital computer with established linearization techniques in such a fashion that it can be used as a method of identifying parameters. The mathematics used in the program is developed in detail and an example is given of its use.

Essentially the method is an efficient device of searching for unknown parameters existing in a set of algebraic or differential equations. The mathematical concepts are historical but the combination of historical mathematics with the high-speed digital computer yields new and useful results.

TABLE OF CONTENTS

1. GENERAL DESCRIPTION

In many computer applications there is a set of test data Y_{ji} (value of some given function g at a time sample t_i) from an experiment and a mathematical model of the experiment that takes the form,

$$g_j(x(t)) \qquad j = 1, NK \tag{1.1}$$

where $x(t)$ is constrained by N first order differential equations.

$$\dot{x}(t) = f(x(t),t;c) \tag{1.2}$$

with initial conditions $x(t_o)$. The vectors $x(t)$, $f(x(t),t;c)$ and c are

$$x(t) \equiv (x_1(t),x_2(t),\ldots,x_n(t)) \tag{1.3}$$

$$f(x(t),t;c) \equiv (f_1(x(t),t;c),f_2(x(t),t;c),\ldots,f_n(x(t),t;c)) \tag{1.4}$$

$$c \equiv (c_1, c_2,\ldots,c_L, c_{L+1}, c_{L+2},\ldots,c_{L+K}), \; L \leq N \tag{1.5}$$

The first L components of the vector c represents a set of L unknown initial conditions and the $L+1$ to $L+K$ components of c are unknown parameters of a given set of differential equations. Determining the vector c from test data would provide a complete mathematical model of the experiment.

One method to determine c would be to select it in some sort of intuitive way, substitute it into (1.2), integrate (1.2), substitute this value of $x(t)$ into (1.1) and then compare with the known test data Y_{ji}. From the results an intuitive guess could be made for another c that might bring $g_j(x(t_i))$ closer to the test data Y_{ji}. A search of this type is time consuming and requires guessing as one approaches the concept of a "good fit."

Richard Bellman [1] has described a method that uses the computer to search for a vector c that minimizes the equation.

$$S(c) = \sum_{i=1}^{NP} \sum_{j=1}^{NK} \lambda_j (g_j(x(t_i)) - Y_{ji})^2 \tag{1.6}$$

where $x(t)$ is determined from $\dot{x}(t) = f(x(t),t;c)$ with initial conditions $x(t_o)$ and λ_i are relative weights of NK different functions $e_i(x(t))$.

This report describes the theory used in the CAL computer program that searches for the minimum of equation (1.6). The program was originally obtained at UCLA and then modified extensively by Dr. John T. Fleck to improve the convergence techniques for a given solution.

2. MATHEMATICAL ANALYSIS

2.1 Approximations and Assumptions

In equation (1.5) the components of vector c are assumed to be composed of L unknown initial conditions and K unknown parameters.

Although the equation $g_j(x(t))$ in many cases will be simply $g_j(x(t)) = x_j(t)$, for the purpose of being general $g_j(x(t))$ is considered as any function of $x(t)$ that may be approximated by,

$$(2.1.1) \qquad g_j(x(t)) = g_j(x(t)_o) + \sum_{k=1}^{N} (x_k(t) - x_k(t)_o) \frac{\partial g_j(x(t)_o)}{\partial x_k}$$

where $x(t)_o$ is the vector generated by assuming (or calculating) an initial vector c, (1.5), and integrating (1.2) over the desired range of the independent variable t. It is about this nominal vector $x(t)_o$ that the linearization (2.1.1) takes place [(1)].

Substituting (2.1.1) into (1.6) gives,

$$S(c) = \sum_{i=1}^{NP} \sum_{j=1}^{NK} \lambda_j \left[g_j(x(t_i)_o) + \sum_{k=1}^{N} (x_k(t_i) - x_k(t_i)_o) \frac{\partial g_j(x_i(t)_o)}{\partial x_k} - Y_{ji}\right]^2$$

(2.1.2)

where

$x_k(t)_o$ = the nominal or present k-th component of the vector $x(t)$.

$x_k(t)$ = the improved k-th component of vector $x(t)$.

Let

$$(2.1.3) \qquad x_k(t) = x_k(t)_o + \epsilon_k(t)$$

where

$\epsilon_k(t)$ = a correction on the vector component $x_k(t)$.

---

(1)

$x(t)_o$ is the nominal or present value of vector $x(t)$.

$x(t_i)$ is the value of the vector $x(t)$ at $t = t_i$.

Assume that $\epsilon_k(t)$ may be approximated by the linear expression,

$$\text{(2.1.4)} \qquad \epsilon_k(t) = \sum_{j=1}^{L} \delta_j h_{kj}(t) + \sum_{j=1}^{K} \gamma_j \, r_{kj}(t) \qquad k = 1, N$$

where

δ_j = corrections to unknown initial conditions

γ_j = corrections to the unknown parameters

Thus,

$$\text{(2.1.5)} \qquad x_j(t_o)_{\text{updated}} = x_j(t_o)_{o \ \text{present guess}} + \delta_j, \qquad 1 \leq j \leq L$$ [2]

$$\text{(2.1.6)} \qquad c_{j \ \text{updated}} = c_{jo \ \text{present guess}} + \gamma_{j-L}, \qquad L+1 \leq j \leq L+K$$

and the functions $h_{ki}(t)$ and $r_{ki}(t)$ are solutions to the linear differential equations [1].

$$\text{(2.1.7)} \qquad \dot{h}_{ki}(t) = \sum_{n=1}^{N} \frac{\partial f_k}{\partial x_n} h_{ni}(t), \qquad i = 1, L$$ [3]

where $h_{nj}(t_o) = \delta_n^j$ [4]

$$\text{(2.1.8)} \qquad \dot{r}_{ki} = \sum_{n=1}^{N} \frac{\partial f_k}{\partial x_n} r_{ni}(t) + \frac{\partial f_k}{\partial c_{L+i}} \qquad \begin{matrix} j = 1, K \\ k = 1, N \end{matrix}$$

where $r_{nj}(t_o) = 0$.

It is assumed that $x(t)$ can be linearized so that to a first approximation the corrections $\epsilon_k(t)$ depend linearly on δ_i and γ_i

(1) Where these differential equations and their initial conditions come from will be shown in Section 2.2.

(2) $c_j \equiv x_j(t_o)$ $\qquad c_{jo} \equiv x_j(t_o)_o$

(3) $f_k \equiv f_k(x(t), t; c)$ here and in the sequel.

(4) $\delta_n^i = 1$ for $n = i$

$= 0$ otherwise

and when convergence is obtained

$$(2.1.9) \qquad \delta_i \longrightarrow 0, \qquad \gamma_j \longrightarrow 0, \text{ and } \qquad \epsilon_k(t) \longrightarrow 0$$

2.2 Linearization of the Differential Equations

Linearization of the differential system (1.2) yields equations (2.1.7) and (2.1.8). To show how these differential equations are derived, rewrite for convenience equation (1.2),

$$(2.2.1) \qquad \dot{x}(t) = f(x(t), t; c)$$

where the initial conditions,

$$(2.2.2) \qquad x_i(t_o) \approx x_{io} \equiv c_i \text{ are given initial condition guesses}$$

$$i = 1, I \, .$$

and

$$(2.2.3) \qquad c_i \approx c_{io} \qquad \text{are given parameter guesses}$$

$$i = I + 1, I + K \, .$$

Known initial conditions, if any, are,

$$(2.2.4) \qquad x_j(t_o) = x_{jo}, \qquad j = I + 1, N \, .$$

Using the guessed initial conditions (2.2.2), the guessed parameters (2.2.3) and the known initial conditions (2.2.4) integrate (2.2.1) from $t = t_o$ to $t = T$ (Final Time) and store the solution at i equally spaced time points $t = t_i$. Designate this solution as $x(t)_o$.

Linearizing equations (2.2.1) about $x(t)_o$ and c_o gives,

$$(2.2.5) \quad \dot{x}(t) \approx f(x(t)_o, t, c_o) + \sum_{n=1}^{N} \frac{\partial f_i}{\partial x_n}(x(t) - x(t)_o) + \sum_{n=I+1}^{I+K} \frac{\partial f_i}{\partial c_n}(c_n - c_{no})$$

Substitution of equations (2.1.3) and (2.1.6) into (2.2.5) gives,

$$(2.2.6)\quad \dot{x}_k(t)_o + \dot{\epsilon}_k(t) \approx f_k(x(t)_o) + \sum_{n=1}^{N} \frac{\partial f_k}{\partial x_n}\,\epsilon_n(t) + \sum_{n=1}^{K} \frac{\partial f_k}{\partial c_{L+n}}\,\gamma_n$$

Substituting (2.1.4) into (2.2.6) and collecting terms of δ_i and γ_j gives,

$$(2.2.7)\quad \sum_{j=1}^{L} \delta_j \left[\dot{h}_{kj}(t) - \sum_{n=1}^{N} \frac{\partial f_k}{\partial x_n}\, h_{nj}(t)\right] + \sum_{j=1}^{K} \gamma_j \left[\dot{r}_{kj}(t) - \sum_{n=1}^{N} \frac{\partial f_k}{\partial x_n}\, r_{nj}(t) - \frac{\partial f_k}{\partial c_{L+j}}\right] = 0$$

Equating the coefficients of δ_j and γ_j equal to zero gives the differential equations (2.1.7) and (2.1.8).

For convenience of notation δ_j, γ_j, $h_{kj}(t)$ and $r_{kj}(t)$ will be redefined as,

(2.2.8)	$d_j = \delta_j$	$j = 1, L$	
(2.2.9)	$d_{j+L} = \gamma_j$	$j = 1, K$	
(2.2.10)	$W_{kj}(t) = h_{kj}(t)$	$j = 1, L,$	$k = 1, N$
(2.2.11)	$W_{kj}(t) = r_{kj}(t)$	$j = L+1, K,$	$k = 1, N$

Equations (2.2.8) - (2.2.11) may now be used to define equation (2.1.4) as,

$$(2.2.12)\quad \epsilon_k(t) = \sum_{j=1}^{LK=L+K} d_j\, W_{kj}(t) \qquad k = 1, N$$

and the differential set of equations (2.1.7) and (2.1.8) becomes,

$$(2.2.13)\quad \dot{W}_{kj} = \sum_{n=1}^{N} \frac{\partial f_k}{\partial x_n}\, W_{nj}, \qquad j = 1, L \qquad k = 1, N$$

(2.2.14) $$\dot{W}_{kj} = \sum \frac{\partial f_k}{\partial x_n} W_{nj} + \frac{\partial f_k}{\partial c_j}, \quad j = L + 1, K, \quad k = 1, N$$

To establish the initial conditions for equations (2.2.13) and (2.2.14) note that at $t = t_o$, equation (2.1.3) is,

(2.2.15) $$x_k(t_o) = x_k(t_o)_o + \epsilon_k(t_o), \qquad k = 1, L$$

and equation (2.1.5) is,

(2.2.16) $$x_k(t_o) = x_k(t_o)_o + \delta_k, \qquad k = 1, L$$

which implies that at $t = t_o$, $\epsilon_k(t_o) = \delta_k$.

Since $\epsilon_k(t_o) = \delta_k$ at $t = t_o$, the initial conditions for the set of linearized differential equations (2.1.7) and (2.1.8) are:

(2.2.17) $$W_{kj}(t_o) = \delta_k^j \qquad j = 1, L \qquad k = 1, N$$

(2.2.18) $$W_{kj}(t_o) = 0 \qquad j = L + 1, K \qquad k = 1, N$$

2.3 Solving For Corrections to Unknown Initial Conditions and Unknown Parameters.

Using equations (2.1.3) and (2.2.12), equation (2.1.2) may be expressed as,

(2.3.1) $$S(d) = \sum_{i=1}^{NP} \sum_{j=1}^{NK} \lambda_j \left[g_j(x(t_i))_o + \sum_{m=1}^{LK} d_m \sum_{k=1}^{N} W_{km}(t_i) \frac{\partial g_j(x(t_i))_o}{\partial x_k} - Y_{ji} \right]^2$$

or,

(2.3.2) $$S(d) = \sum_{i=1}^{NP} \sum_{j=1}^{NK} \lambda_j \left[\sum_{m=1}^{LK} d_m \sum_{k=1}^{N} W_{km}(t_i) \frac{\partial g_j(x(t_i))_o}{\partial x_k} - Y_{ji} + g_j(x(t_i))_o \right]^2$$

Setting,

$$(2.3.3)\qquad g_{jm}(t_i) \equiv \sum_{k=1}^{N} W_{km}(t_i)\,\frac{\partial g_j(x(t_i))_o}{\partial x_k}$$

(2.3.2) becomes,

$$(2.3.4)\qquad S(d) = \sum_{i=1}^{NP}\sum_{j=1}^{NK}\lambda_j\left[\sum_{m=1}^{LK} d_m\, g_{jm}(t_i) - Y_{ji} - g_j(x(t_i)_o)\right]^2$$

The set of d_m's that make (2.3.4) a minimum are determined by taking the partial of S(d) with respect to d_m and **setting these partials** to zero. Thus,

$$\frac{\partial S(d)}{\partial d_k} = 2\sum_{i=1}^{NP}\left[\sum_{i=1}^{NK}\lambda_j\sum_{m=1}^{LK} d_m\, g_{jm}(t_i) - (Y_{ji} - g_j(x(t_i)_o))\right] g_{jk}(t_i) = 0$$

(2.3.5) $k = 1, LK$

or,

$$\sum_{m=1}^{LK} d_m\left[\sum_{i=1}^{NP}\sum_{i=1}^{NK}\lambda_j\, g_{jm}(t_i)\, g_{jk}(t_i)\right] = \sum_{i=1}^{NP}\sum_{i=1}^{NK}\lambda_j (Y_{ji} - g_j(x(t_i)o)\, g_{jk}(t_i))$$

$$= \sum_{m=1}^{LK} E_{km}\, d_m = D_k \qquad k = 1, LK$$

(2.3.6)

where

$$(2.3.7)\qquad E_{km} = \sum_{i=1}^{NP}\sum_{j=1}^{NK}\lambda_j\, g_{jm}(t_i)\, g_{jk}(t_i)$$

$$D_k = \sum_{i=1}^{NP}\sum_{j=1}^{NK}\lambda_j\left[Y_{ji} - g_j(x(t_i)o)\, g_{jk}(t_i)\right] \qquad k = 1, LK$$

Solving the linear set of algebraic equations (2.3.6) in the LK unknowns d_m gives the corrections to the initial conditions and parameters.

Thus, the updated estimates will be,

$$x_j(t_o) = x_j(t_o)_o + d_j, \qquad j = 1, L \tag{2.3.8}$$

$$c_j = c_{jo} + d_j, \qquad j = L + 1, LK \tag{2.3.9}$$

3. COMPUTER FLOW DIAGRAM

In Figure 1 a very general computer flow diagram is represented. The purpose of this diagram is not to explain all the details of the program but to give a general idea how the equations developed in Section 2 are programmed. A brief description of the seven blocks shown in Figure 1 is as follows:

1. Read Input Data requires that the user furnish an initial guess for the vector c or a subroutine that produces a set of initial guesses from the data (equation (1.5)). This amounts to furnishing L initial guesses to the unknown initial conditions and K initial guesses to the unknown parameters. Known initial conditions, system constants, and test data must also be furnished.

2. Solving equation (1.2) for a complete solution over the interval (t_0, T) is performed by a numerical integration scheme that is part of our present CAL computer facility. The results of this integration are stored temporarily in core. Symbolically, this solution is called $x(t)_o$ (It is important to note th . $x(t)_o \neq x(t_o)$).

3. Solve linearized equations and accumulate elements of matrices E and D.

4. Solves a set of L + K linear equations given by (2.3.6) for the L + K unknowns d_i. The d_i, i = 1, L + K are corrections to the L unknown initial conditions and the K unknown parameters.

5. Correct the initial guess for vector c by,

$$c_i = c_i + d_i \qquad i = 1, L$$

for initial conditions

and,

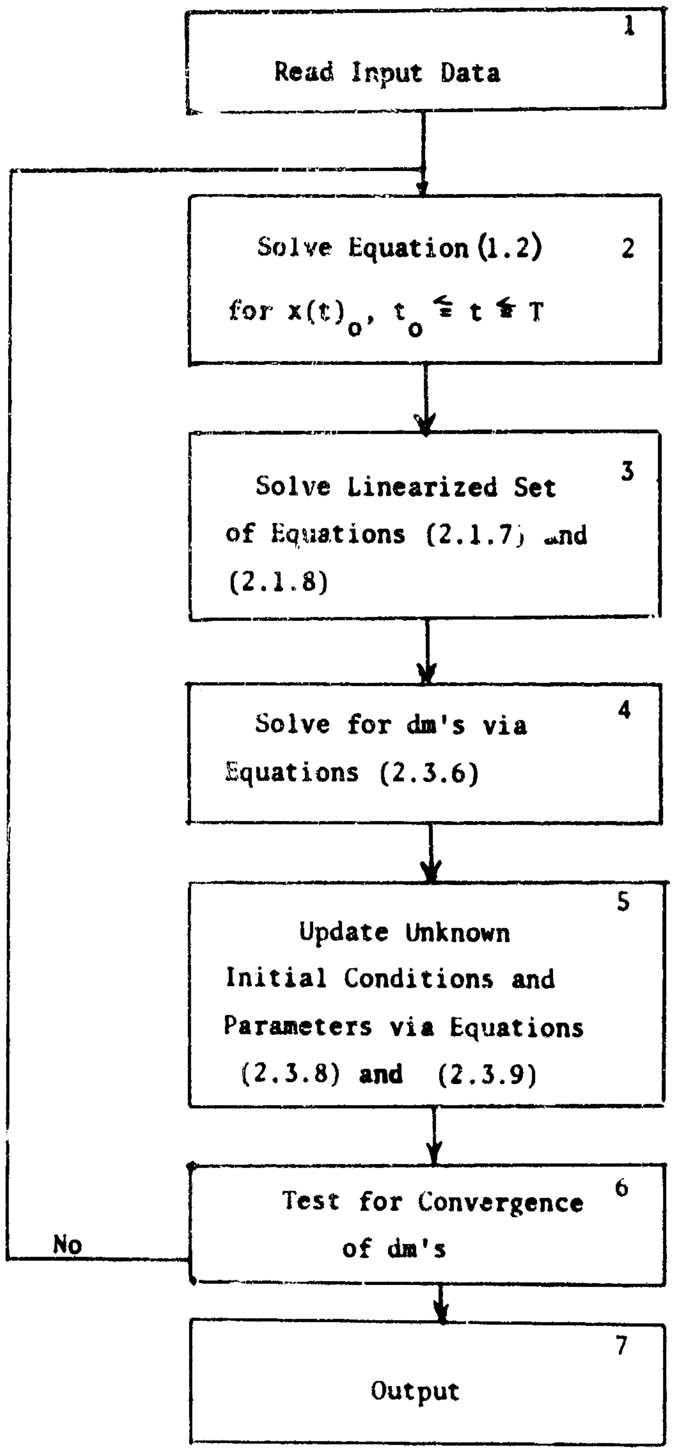
$$c_i = c_i + d_i \qquad i = L + 1, L + K \quad ,$$

for the parameters.

6. A sufficient condition for convergence of the vector c is $|\Delta c_i| / |c_i| \leq \epsilon$ for $i = 1, L + K$. ϵ is a parameter chosen by the user. Special convergence tests and modifications of the d_i are included to prevent large corrections from disturbing the solution.

7. Output is generated in a subroutine in the form of tables and/or graphs.

The success of the program was due largely to the inclusion of techniques for handling subsets of the given data which were expanded to include the complete data as the progress converged.



Computer Flow
Diagram

Figure 1

4. IDENTIFICATION OF PARAMETERS EXAMPLE

The following problem of Aircraft Parameter Identification is used here to illustrate the use of results developed in Section 2. For this problem equation (1.1) is,

$$
\begin{aligned}
g_1(x(t)) &\equiv x_1(t) \\
g_2(x(t)) &\equiv x_2(t) \\
g_3(x(t)) &\equiv x_3(t) \\
g_4(x(t)) &\equiv x_4(t)
\end{aligned}
\tag{4.1}
$$

Where $x_1(t)$, $x_2(t)$, $x_3(t)$ and $x_4(t)$ are constrained by the set of differential equations,[1]

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}
=
\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & c_1 & c_2 & c_3 \\ .054\text{[illegible]}7 & c_5 & c_6 & -1 \\ 0 & c_8 & c_9 & c_{10} \end{bmatrix}
\times
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
+
\begin{bmatrix} 0 \\ c_4 \\ c_7 \\ c_{11} \end{bmatrix}
\tag{4.2}
$$

For this problem it is assumed that all initial conditions are known and are:

$$
x_j(t_o) = 0, \qquad j = 1, 4 \tag{4.3}
$$

Thus, there are no unknown initial conditions so that $L = 0$ in the identity (1.5). In (4.2) there are 11 unknown parameters (c_i, $i = 1, 11$) so that $K = 11$ in identity (1.5). The problem then is to find a vector $c = (c_1, c_1, \ldots c_{11})$ that minimizes (1.6). The program does this provided an initial guess is assumed or calculated. For this problem, the initial guess for the vector c was calculated by using spline functions [2].

4.1 Estimating Initial Vector c By Spline Functions

Test data (designated by Y_{ji} ($j = 1, 4$; $i = 1, 95$) was given for

---

(1) $x_i = x_i(t)$, $i = 1, 4$ here and in the sequel.

$x_1(t_i)$, $x_2(t_i)$, $x_3(t_i)$ and $x_4(t_i)$ at 95 equally spaced points. Using this data spline functions were calculated. Substituting these splined functions (designated by $\hat{x}_1(t)$, $\hat{x}_2(t)$, $\hat{x}_3(t)$ and $\hat{x}_4(t)$) into equations (4.2) and integrating gives the set of estimates.

$$\begin{bmatrix} \tilde{x}_1(t_i) \\ \tilde{x}_2(t_i) \\ \tilde{x}_3(t_i) \\ \tilde{x}_4(t_i) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & c_1 & c_2 & c_3 \\ .05467 & c_5 & c_6 & -1 \\ 0 & c_8 & c_9 & c_{10} \end{bmatrix} \begin{bmatrix} \int_o^{t_i} \hat{x}_1 \, dt \\ \int_o^{t_i} \hat{x}_2 \, dt \\ \int_o^{t_i} \hat{x}_3 \, dt \\ \int_o^{t_i} \hat{x}_4 \, dt \end{bmatrix} + t_i \begin{bmatrix} 0 \\ c_4 \\ c_7 \\ c_{11} \end{bmatrix}$$

(4.4) $i = 1, 95$

Forming a measure of the differences between Y_{ji} and $\tilde{x}_j(t_i)$ squared as,

(4.5) $$E = \sum_{j=1}^{4} \sum_{i=1}^{95} \frac{1}{2} (\tilde{x}_j(t_i) - Y_{ji})^2$$

gives for

(4.6) $$\frac{\partial E}{\partial c_k} = 0 \qquad k = 1, 11$$

a set of eleven linear equations in the unknown c_k, $k = 1, 11$. This method proved to be an effective way to get an initial estimate for vector c.

4.2 Equations Users Need To Furnish

The user needs to provide the mathematical model of the experiment such as is given by equations (1.1) and their partials with respect to x_k. Further partials of equations (1.4) with respect to x_k and c_k are needed. Input data, system constants, and convergence test constants are also needed.

4.3 Results

Figures 1 through 4 are the results of integrating equations (1.2)

(after convergence of the vector c). The symbol x in the figures indicates the Y_{ji} data points for j = 1, 4 and i = 1, 95 and the continuous curves represent the final solutions of equations (1.2). The vector components of c given on Figure 1 is the set of parameters that this quasilinearization program converged to and this vector c represents a local minimum of the function S(c) given by equation (1.6).

Some interesting things to note are that:

1. Equation (1.2) can be non-linear.
2. Initial guesses for c_k's are not needed if equations (1) are linear in c_k's (linear or non-linear in x). This was the case described in 4.1.
3. If equation (1.2) is non-linear in c_k's, a set of initial guesses of the c_k's is required. (In many cases slight changes in the math model will permit one to estimate c_k's).
4. The program finds a local minimum of (1.6). Searching techniques would have to be derived to find a global minimum.

It is the opinion of the authors that many estimation problems can be solved by this method of quasilinearization. Simplicity of application and speed (about 12 minutes to solve the above example) appears to be some of its values. Combining the use of the high speed computer and historical mathematics gives onethe tools to solve a host of identification problems that in the past would have demanded many hours of a human guided search technique to converge on a local minimum solution.

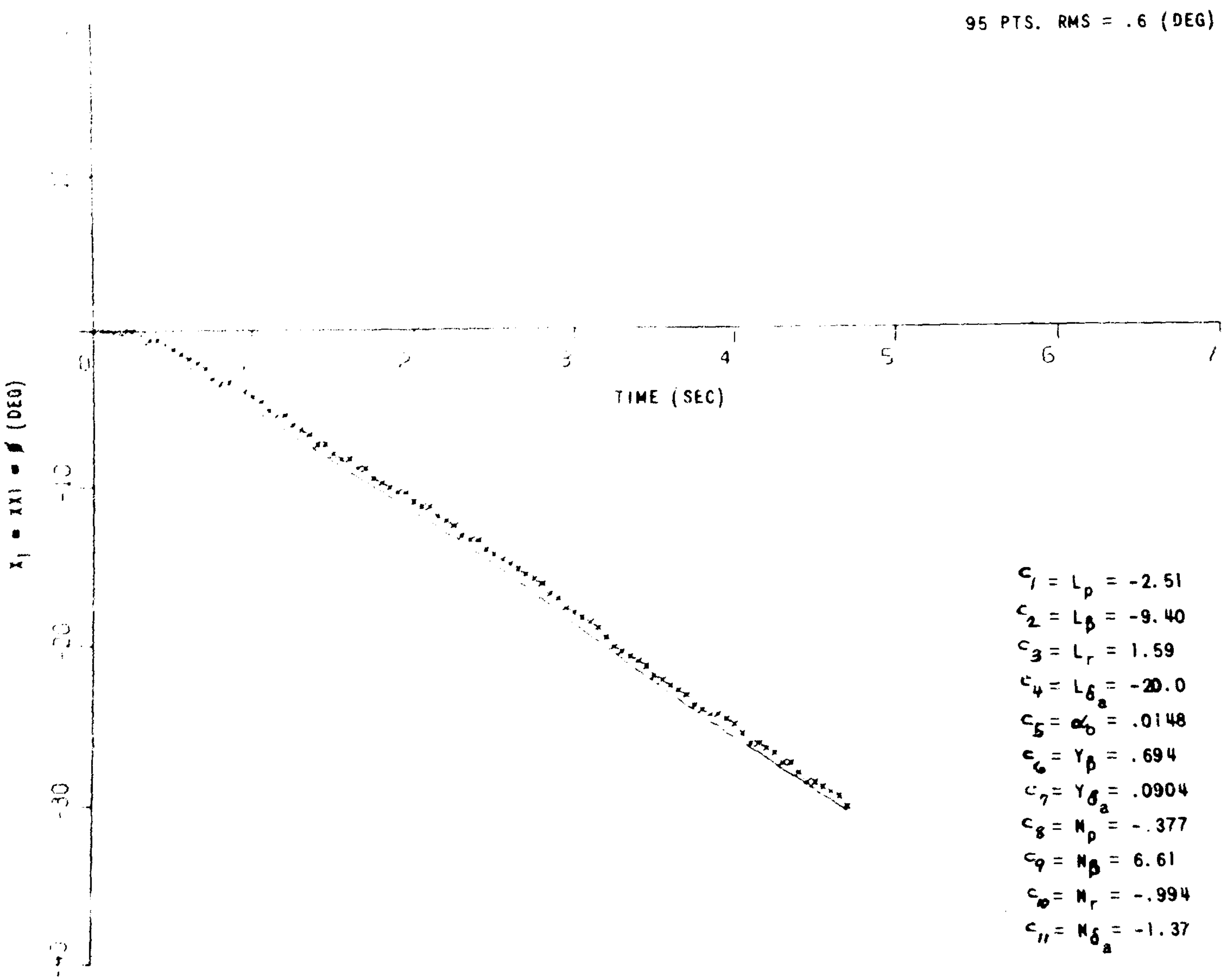


Figure 1 CONVERGED ESTIMATE OF STATE X_1

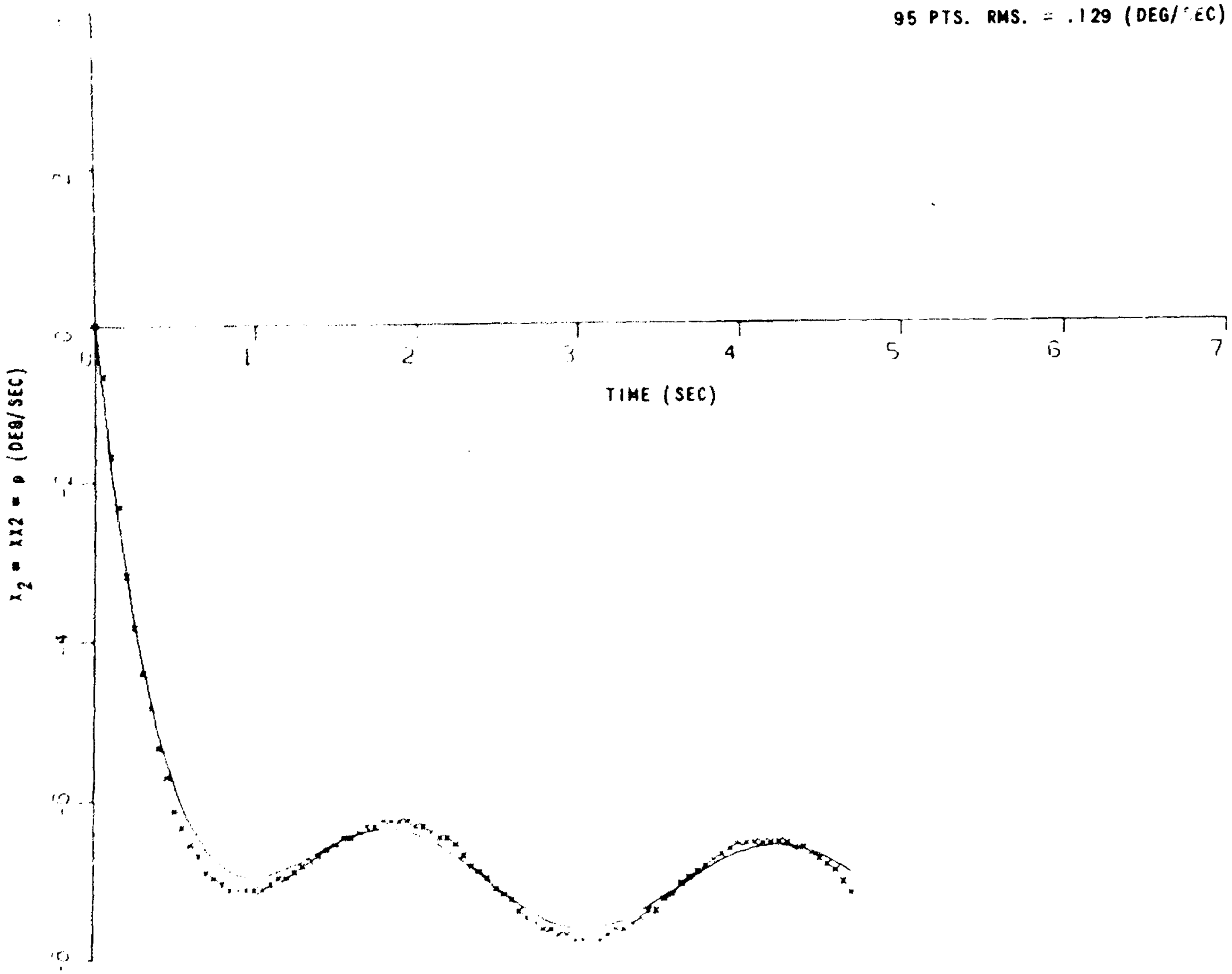


Figure 2 CONVERGED ESTIMATE OF STATE X_2

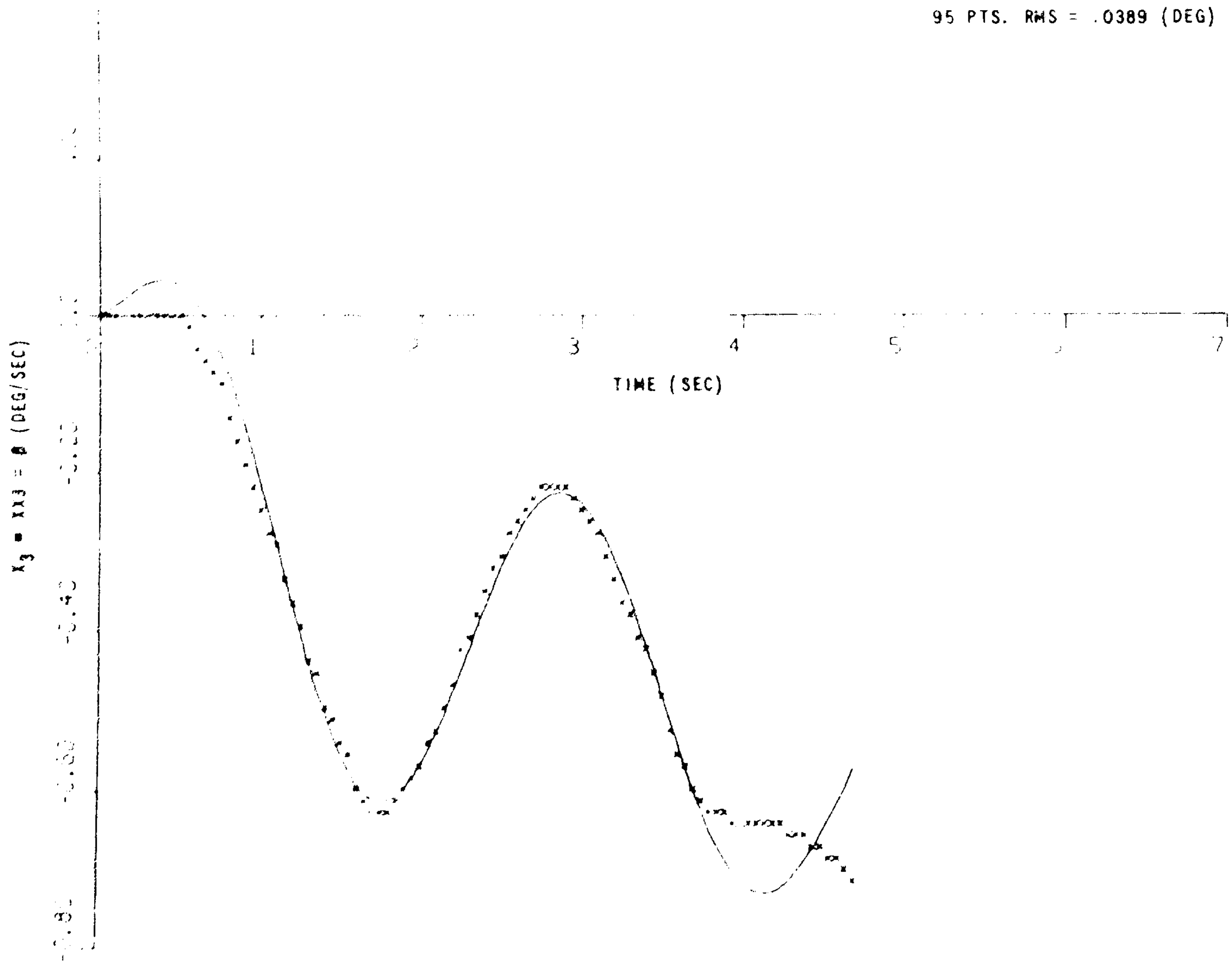


Figure 3 CONVERGED ESTIMATE OF STATE X_3

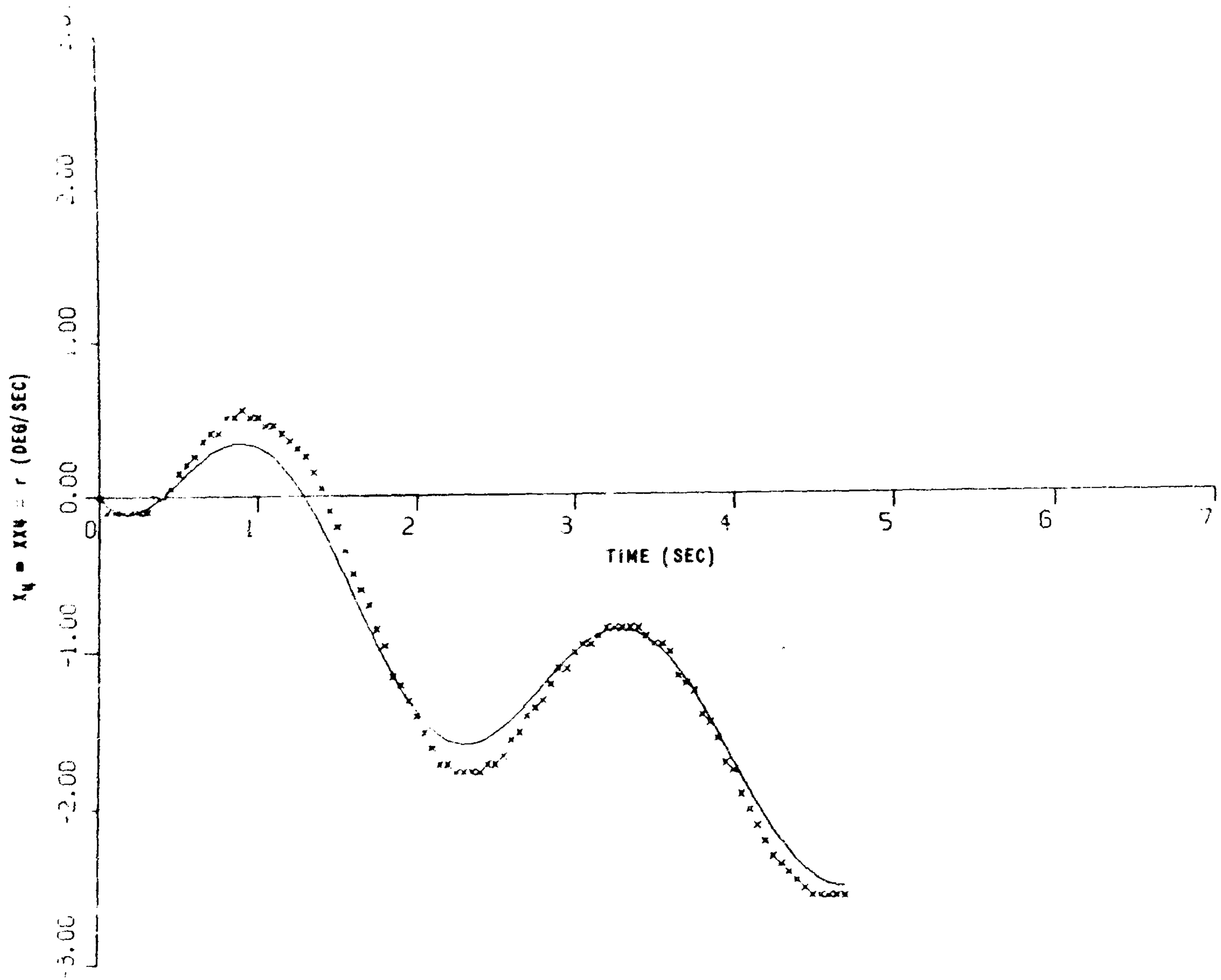


Figure 4 CONVERGED ESTIMATE OF STATE X_4

REFERENCES

1. Bellman, R. F. Kalaba, R. E., Quasilinearization and Non-linear Boundary-Value Problems, American Elsevier Publishing Company, Inc., New York, 1965.

2. Ahlberg, J. H., Nilson, E. N., Walsh, J. L., The Theory of Splines and Their Applications, Academic Press, New York, 1967.